AIAA SciTech Forum
11–15 & 19–21 January 2021, VIRTUAL EVENT
AIAA Scitech 2021 Forum

10.2514/6.2021-1939

High-order discontinuous Galerkin Difference cut-cell discretization

Sharanjeet Kaur * and Jason E. Hicken †
Rensselaer Polytechnic Institute, Troy, New York-12180

We present a high-order discontinuous Galerkin difference (DGD) discretization for cut-cell meshes. We leverage the DGD reconstruction procedure to ameliorate the small-cell problem associated with cut-cells. Numerical experiments demonstrate that the conditioning of DGD discretizations is insensitive to cut-cell sizes for linear problems in one- and two-dimensions. In addition, results are presented that verify the accuracy of the DGD discretization applied to the Euler equations.

I. Introduction

Computational Fluid Dynamics (CFD) has become an increasingly useful and powerful tool for aerospace engineers. For example, high-fidelity CFD simulations are now routinely combined with optimization algorithms for design. Nevertheless, the automation of CFD-based design optimization remains challenging, because human expertise remains a common component of the mesh generation process.

The design optimization process can be automated in the case of simple geometries, for which grid generation does not require user input or mesh movement does not introduce significant errors. However, in the case for complex geometries, or geometries where the flow physics change significantly during the optimization, automation remains challenging for conventional grid-generation workflows. Grid generation is especially difficult for high-order, conforming meshes, in which boundary elements must be curved to match the surface. This has hindered the adoption of high-order methods in industry, in general, and in design optimization, in particular.

Cut-cell methods can be used to enable high-order mesh generation for design optimization, since they offer the potential to make the flow simulations around complex, curved geometries more robust and automated compared to body-fitted meshes. Cut-cell methods have been used in many applications, including the full potential equations [1], the Euler equations [2–7], the Navier-Stokes equations [8, 9], and the wave equation [10].

In cut-cell methods, the mesh often (but not always) begins with a Cartesian grid that does not conform to the underlying geometry. The elements that intersect with the embedded geometry's boundary are referred to as cut-cells. Only the portion of the cut-cells lying inside the domain of interest are considered in the discretization, so these methods must handle cut-cells of arbitrary shape. While cut-cell methods are typically applied to square/cube elements, they have also been applied to triangular [9] and tetrahedral elements [11]. In the finite-element literature, the cut-cell method is also known as CutFEM [12, 13].

Along with its potential benefits, there are also some difficulties with using cut cells. One of the challenges encountered by this class of methods is the so-called small-cell problem. A cut-cell can be arbitrarily small depending on how it intersects with the geometry, and these small cells may produce ill-conditioned Jacobians or unstable discretizations [12]. Consequently, small cells may require severe time-step restrictions when an explicit time-marching method is used [7], or a significant number of inner iterations when an implicit time marching method is used.

The small-cell problem has been addressed in several distinct ways, including cell merging [2, 3, 7, 11, 14, 15], in which small cells are merged with larger neighbors; a wave propagation technique [4], in which the size of the stencil at, or near, small cells is increased to maintain stability, and; stabilization [12, 16, 17], in which penalty terms are introduced to couple the degrees of freedom in small cells with neighboring cells.

Cell merging, in particular, has been popular among different cut-cell methods to deal with the small-cell problem; however, one of the problems faced by this method while dealing with the small-cell problem is that it may not always be straightforward to decide on which cells should be considered "small".

Specialized preconditioners have also been developed as a solution to the small-cell problem. For example, in [18] the authors propose an optimal preconditioner in the sense that the resulting condition number is independent of the

*Graduate Student, Department of Mechanical, Aerospace and Nuclear Engineering
†Associate Professor, Department of Mechanical, Aerospace and Nuclear Engineering, Senior Member AIAA

mesh size and the embedded boundary position for elliptic problems. However to the best of our knowledge, such a preconditioner has not been developed for advection-dominated problems. Furthermore, preconditioning is not a suitable solution to a restrictive CFL condition when using an explicit time integration scheme.

In this paper, we investigate a cut-cell version of the discontinuous Galerkin difference (DGD) method [19–21] that does not appear to suffer from the small-cell problem. DGD discretizations are based on piecewise discontinuous polynomial basis functions whose support extends over several elements; we believe the stencil of the DGD reconstruction eliminates the small-cell problem without requiring special treatment. This claim is supported by the preliminary results presented herein.

The rest of the paper is organized as follows. Section II presents the governing equations and their weak formulation; the section also describes the level-set method used to represent the geometry and develop quadratures for cut-cells. Section III provides the cut-cell DGD discretization for the Euler equations, and accuracy studies are presented in Section IV. Finally, we conclude with a summary in Section V.

II. Cut-cell finite-element formulation using level-sets

This section provides a high-level overview of how the cut-cell finite-element method can be used to discretize the Euler equations of gas dynamics. Consider the strong form of the two-dimensional Euler equations:

$$\frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} + \frac{\partial \boldsymbol{\mathcal{F}}_x}{\partial x} + \frac{\partial \boldsymbol{\mathcal{F}}_y}{\partial y} = \mathbf{0}, \qquad \forall\, \boldsymbol{x} \in \Omega, \tag{1}$$

where the state variables are the conservative variables, $\boldsymbol{\mathcal{U}} = [\rho, \rho u, \rho v, e]^T$, and the flux vectors are

$$\boldsymbol{\mathcal{F}}_x = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e+p)u \end{bmatrix}, \qquad \boldsymbol{\mathcal{F}}_y = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ (e+p)v \end{bmatrix}.$$

The pressure is defined by the calorically perfect ideal gas law as $p = (\gamma - 1)[e - \frac{\rho}{2}(u^2 + v^2)]$, with $\gamma = 1.4$. For simplicity, we do not include boundary conditions for the time being.

We begin the discretization process by partitioning the domain into K elements of uniform size*; that is, $\Omega = \bigcup_{\kappa=1}^{K} \Omega_\kappa$ where Ω_κ denotes the domain of a single element κ with boundary Γ_κ. Let $h = \max_\kappa \sqrt{\Omega_\kappa}$ be the nominal element size (in 2D). To obtain the Galerkin weak formulation of the Euler equations, we multiply (1) by a test function and integrate over each element to yield the following problem statement: find $\boldsymbol{\mathcal{U}} \in W(\Omega)$ such that

$$\sum_{\kappa=1}^{K} \int_{\Omega_\kappa} \mathcal{V} \frac{\partial \boldsymbol{\mathcal{U}}}{\partial t}\, d\Omega_\kappa - \sum_{\kappa=1}^{K} \int_{\Omega_\kappa} \left(\frac{\partial \mathcal{V}}{\partial x} \boldsymbol{\mathcal{F}}_x + \frac{\partial \mathcal{V}}{\partial y} \boldsymbol{\mathcal{F}}_y \right) d\Omega_\kappa + \sum_{\kappa=1}^{K} \oint_{\Gamma_\kappa} \mathcal{V} \left(\boldsymbol{\mathcal{F}}_x n_x + \boldsymbol{\mathcal{F}}_y n_y \right) d\Gamma_\kappa = 0, \qquad \forall\, \mathcal{V} \in W(\Omega), \tag{2}$$

where $W(\Omega)$ is an appropriate function space. Note that we have used integration-by-parts on each element to arrive at the weak form (2).

The infinite-dimensional problem (2) is transformed into a finite-dimensional problem by replacing $W(\Omega)$ with a finite element space, which we denote by $W_h(\Omega)$. Thus, the generic finite-element problem statement is to find $\boldsymbol{\mathcal{U}}_h \in W_h(\Omega)$ such that

$$\sum_{\kappa=1}^{K} \int_{\Omega_\kappa} \mathcal{V}_h \frac{\partial \boldsymbol{\mathcal{U}}_h}{\partial t}\, d\Omega_\kappa - \sum_{\kappa=1}^{K} \int_{\Omega_\kappa} \left(\frac{\partial \mathcal{V}_h}{\partial x} \boldsymbol{\mathcal{F}}_x + \frac{\partial \mathcal{V}_h}{\partial y} \boldsymbol{\mathcal{F}}_y \right) d\Omega_\kappa + \sum_{\kappa=1}^{K} \oint_{\Gamma_\kappa} \mathcal{V}_h \hat{\mathcal{F}}_n(\boldsymbol{\mathcal{U}}_h^+, \boldsymbol{\mathcal{U}}_h^-)\, d\Gamma_\kappa = 0, \qquad \forall\, \mathcal{V}_h \in W_h(\Omega), \tag{3}$$

where $\hat{\mathcal{F}}_n(\boldsymbol{\mathcal{U}}_h^+, \boldsymbol{\mathcal{U}}_h^-)$ denotes a conservative numerical flux function. The flux function depends on the trace value taken from the interior of the element, $\boldsymbol{\mathcal{U}}_h^+$, and the trace value from the exterior of the element, $\boldsymbol{\mathcal{U}}_h^-$; the latter is based on the numerical solution, for interior faces, or the boundary conditions, for boundary faces. The particular numerical flux functions used in this work will be described in Section IV.

*we consider non-conforming, Cartesian adaptive meshes later.

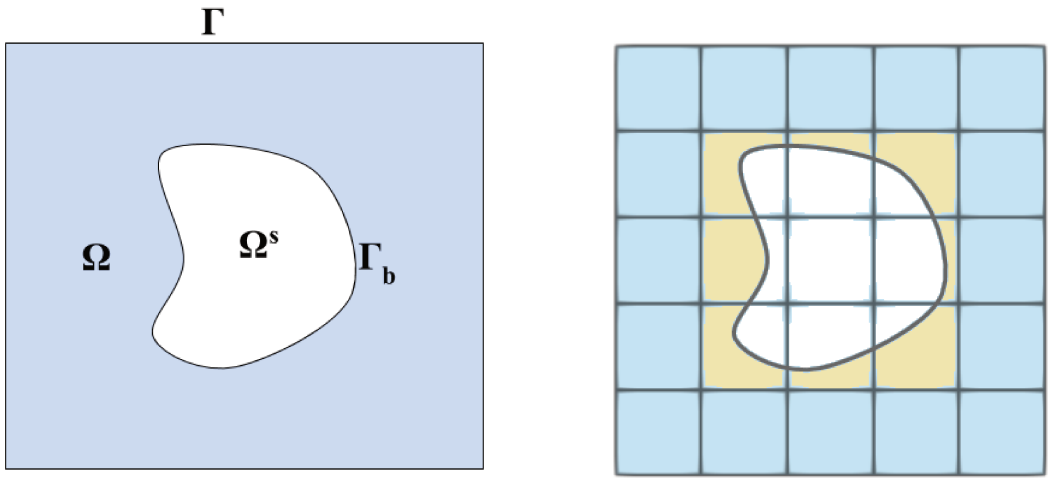
In order to evaluate the integrals in (3), we need a quadrature rule for each element. We can use standard quadrature rules for the elements that are not cut by the boundary; for example, tensor-product Gauss-Legendre quadrature rules for quadrilaterals and hexahedral elements. However, for cut-cells — the elements intersected by the embedded boundary — standard quadrature rules cannot be used, in general. In order to both identify and integrate over cut-cells, we leverage the algorithms in [22], which brings us to the subject of level-sets.

A. Level-set method and cut-cells

Suppose we have a rectangular domain $\bar{\Omega}$ that can be decomposed into a fluid domain, Ω, the immersed boundary, Γ_b, and a solid domain, Ω^s, as shown in Figure 1(a). Let $\bar{\Omega}_\kappa$ be the domain of a single (uncut) element in $\bar{\Omega}$.

We are only interested in the elements that intersect, partially or completely, with Ω in the context of solving (3). For the cut-cells, $\{\Omega_\kappa, \kappa = 1, 2, 3, \ldots, K \mid \bar{\Omega}_\kappa \cap \Gamma_b \neq \emptyset\}$, we only want to integrate over that part of the cell lying in the fluid domain, Ω, as highlighted by the tan-colored regions in Figure 1(b). Consequently, we are left with the problem of (numerically) integrating over cut-cell domains and boundaries of arbitrary shape and size.

Finding suitable numerical quadrature rules for cut-cells is greatly simplified if we represent the geometry as a level-set function [22] and this is the approach adopted in this work. In addition to representing the immersed boundary, level-sets can also be used to quickly identify cut-cells and the elements lying inside/outside of the embedded geometry. Level-set functions have been used in earlier works for this purpose; for example, see [10, 12, 23, 24].



(a) Cartesian mesh domain

(b) Cartesian mesh with cut-cells

Fig. 1 Example flow domain, embedded geometry, and background mesh.

To be more precise, if we wish to identify whether $\boldsymbol{x} \in \bar{\Omega}$ belongs to the solid domain, fluid domain, or immersed boundary, we use a level-set function $\phi : \mathbb{R}^2 \to \mathbb{R}$ as follows:

$$\begin{cases} \phi(\boldsymbol{x}) < 0, & \boldsymbol{x} \in \Omega^s, \\ \phi(\boldsymbol{x}) = 0, & \boldsymbol{x} \in \Gamma_b, \\ \phi(\boldsymbol{x}) > 0, & \boldsymbol{x} \in \Omega, \end{cases} \tag{4}$$

where the boundary of the embedded geometry is given by zero level-set of function $\phi(\boldsymbol{x})$.

B. Quadrature rules for cut cells

Once the cut cells are identified, the next task is to generate an accurate quadrature rule for the corresponding elements. The standard Gauss-Legendre quadrature rules are used for regular (non-cut) quadrilateral elements. For the cut cells, we seek quadrature rules whose points lie strictly inside the portion of element κ that intersects with Ω and,

more importantly, we require the quadrature rules to have strictly positive quadrature weights. The requirement for positive quadrature weights is motivated by our long-term interest in developing an entropy-stable cut-cell discretization. Since we use a level-set function to represent the geometry, we can use the algorithm from Reference [24] to produce quadrature rules for the cut-cells that satisfy our requirements.

The algorithm described in [24] provides high-order accurate quadrature rules to evaluate integrals over surfaces and volumes defined implicitly via a level-set function restricted to a given hyperrectangle. Suppose $\bar{\Omega}_\kappa$ represents a hyperrectangle (mesh element) intersected by the embedded geometry. Then the algorithm presented in [24] provides a quadrature rule that meets our requirements; that is, all quadrature points lie strictly inside their respective domains, $\boldsymbol{x}_i \in \Omega_\kappa = \bar{\Omega}_\kappa \cap \Omega$, and all quadrature weights are strictly positive, $w_i > 0$.

III. Cut-cell DGD discretizations and their conditioning

As discussed in the introduction, one of the numerical challenges associated with cut-cell methods is the so-called small-cell problem. Near the embedded boundary of the domain, some cut cells may be orders of magnitude smaller than the regular (non-cut) mesh cells. This can produce system matrices (or Jacobians) that have eigenvalues with relatively small modulus resulting in poor conditioning [25].

In this work we investigate the impact of the small-cell problem on a discontinuous Galerkin difference (DGD) discretization. Since Galerkin difference (GD) basis functions extend over several elements, we hypothesize that the discretization will automatically ameliorate the small-cell problem. The high-order GD method was originally proposed in the context of continuous basis functions [26], but it was subsequently extended to discontinuous basis functions [19, 20].

We adapt the DGD method of [21] to cut-cell meshes with the following differences: instead of using summation-by-parts (SBP) operators for element-level operations, we use conventional discontinuous Galerkin basis functions; and, instead of triangular elements, we focus on quadrilateral elements. The development of entropy-stable cut-cell operators is a work in progress and will be included in a forthcoming paper.

A. DGD basis functions

The DGD method is a form of finite-element method, albeit based on non-standard basis functions. Therefore, to familiarize readers with the method, we review the discontinuous basis functions used in the one-dimensional formulation of the DGD scheme.

The discrete DGD solution is given by

$$u_h(\boldsymbol{x}) = \sum_{i=1}^{K} u_i \psi_i(\boldsymbol{x}), \tag{5}$$

where u_i is the discrete solution at the center of element i and $\psi_i(\boldsymbol{x})$ is the corresponding discontinuous basis function. The discontinuous basis function in one dimension is given by

$$\psi_i(x) = \begin{cases} \mathcal{P}_{\kappa,i}(x), & \text{if } x_\kappa < x < x_{\kappa+1}, \text{and } i \in S_\kappa, \\ 0, & \text{otherwise,} \end{cases}$$

where S_κ is the stencil of element κ, which is defined later. The function $\mathcal{P}_{\kappa,i}(x) \in \mathbb{P}_p(\Omega_\kappa)$ is the pth order Lagrange interpolant that satisfies the interpolation conditions

$$\mathcal{P}_{\kappa,i}(x_j) = \begin{cases} 1, & x_j = x_i, \\ 0, & \text{otherwise,} \end{cases}$$

where x_j is the center of an element in the stencil S_κ.

The interpolation condition is satisfied for one-dimensional basis functions, but for two- and three-dimensional unstructured grids it will be violated since we only solve the interpolation conditions in a least-squares sense. Nevertheless, the resulting DGD basis functions are still able to exactly represent polynomials of total degree p, provided the stencil S_κ is unisolvent for $\mathbb{P}_p(\Omega_\kappa)$.

The stencil S_κ of an element κ is the set of all degrees of freedom that directly influence the solution on element κ; typically, the stencil S_κ consists of κ itself and some of the neighbouring elements. Starting from κ, elements are added

(a) Non-cut domain

(b) Cut-cell domain

Fig. 2 Domain for 1-D advection

to the stencil S_κ recursively (based on face-adjacent elements) until the stencil is sufficiently large to exactly represent degree p polynomials. For more details, see [21].

B. Role of DGD basis in avoiding the small-cell problem

One of our primary motivations for this work was a hypothesis that DGD would not suffer from the small-cell problem. Our hypothesis is based on the following, informal reasoning. A DGD basis function extends over several elements; consequently, even if an element associated with basis κ is small, the influence of the basis function itself remains significant. In other words, the equation associated with such a basis function is not poorly scaled, and the system remains well conditioned.

We test this hypothesis by investigating the condition number of the stiffness matrix for two linear partial differential equations over a range of cut-cell sizes.

1. One-dimensional advection PDE

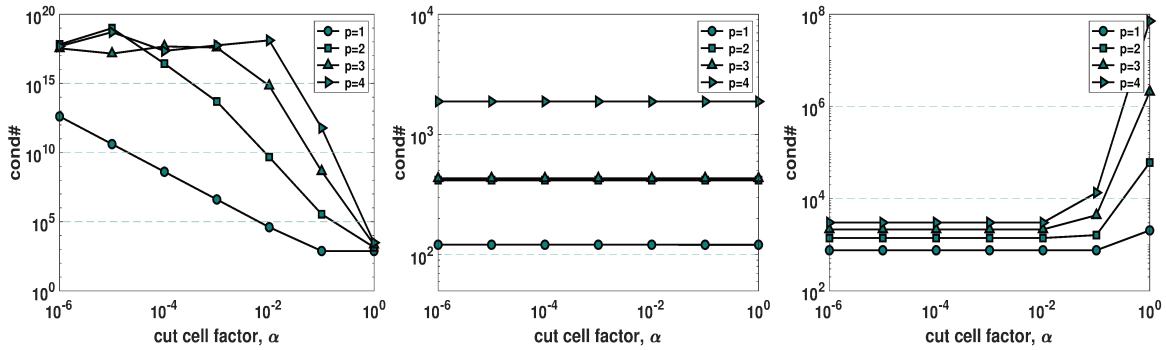
The one-dimensional advection equation (6) is solved using the discontinuous Galerkin (DG) and DGD methods. The domain is cut at the right end as shown in Figure 2(b). Figure 2(a) shows the non-cut domain.

$$\begin{aligned} -a\frac{\partial \mathcal{U}}{\partial \boldsymbol{x}} &= \mathcal{F}, \qquad \forall\, \boldsymbol{x} \in \Omega, \\ \mathcal{U} &= \mathcal{U}_{\mathcal{D}}, \qquad \boldsymbol{x} = 1 - \beta h, \end{aligned} \tag{6}$$

where $\Omega = [0, 1 - \beta h]$ with $\beta = 1 - \alpha$, the advection velocity magnitude is $a = 1$, and the exact solution is $\mathcal{U} = e^x$. The cut-cell size is given by αh with $0 \leq \alpha \leq 1$, where, h is the cell size for a non-cut cell.

In order to study the condition number for different cut-cell sizes, the right boundary cell is cut at different locations; the condition number of the stiffness matrix is then evaluated for each cut-cell size using for both the DG and DGD discretizations. For completeness, we compare with the condition number that results when the cut-cell is merged with its neighbor.

Figure 3 shows the condition number versus cut-cell size for the DG, the cell-merged DG, and the DGD methods. We see that the condition number for DG increases rapidly as the cut-cell size decreases, as shown in Figure 9(a). In contrast, for DGD the condition number remains constant with decreasing cut-cell size; see Figure 9(b). The behavior of DGD is similar to the cell-merged DG case, as shown in Figure 9(c), except for one difference. For cell-merging, as α approaches 1 the condition number grows substantially, so the choice of when to merge a cut-cell with a neighbor is quite critical. In contrast, DGD eliminates this dilemma, since it remains well conditioned for all $\alpha \in [0, 1]$.



(a) DG (b) DGD (c) Cell-merging

Fig. 3 Condition number vs cut-cell factor, α, for a one-dimensional advection PDE

2. Two-dimensional Poisson PDE

Next, we consider the two-dimensional Poisson PDE,

$$-\Delta \mathcal{U} = \mathcal{F}, \qquad \forall\, \boldsymbol{x} \in \Omega, \tag{7}$$

on a square domain, $\Omega = [0,1]^2$, in which a circle with center at $(0.5, 0.5)$ is embedded, as shown in Figure 4. The radius, r, of the circle is varied to obtain cut-cells of different sizes. The PDE is supplied with the following Dirichlet and Neumann boundary conditions;

$$\begin{aligned} \mathcal{U} &= \mathcal{U}_{\mathcal{D}}, \qquad \forall\, \boldsymbol{x} \in \Gamma, \\ \boldsymbol{n}.\nabla \mathcal{U} &= \mathcal{U}_{\mathcal{N}}, \qquad \forall\, \boldsymbol{x} \in \Gamma_b, \end{aligned} \tag{8}$$

where Γ is the outer boundary of the domain and Γ_b is the circular immersed boundary.

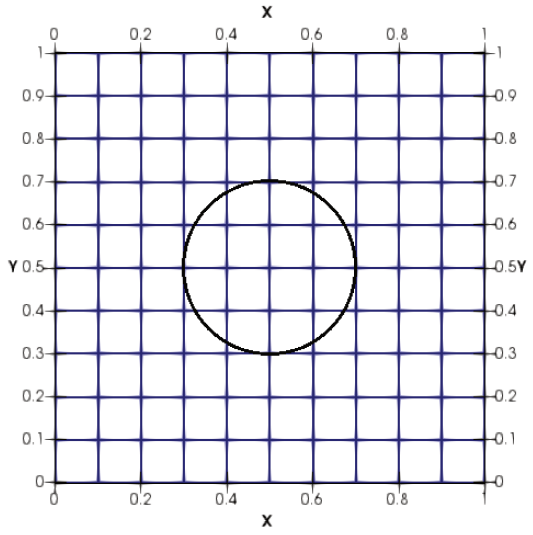


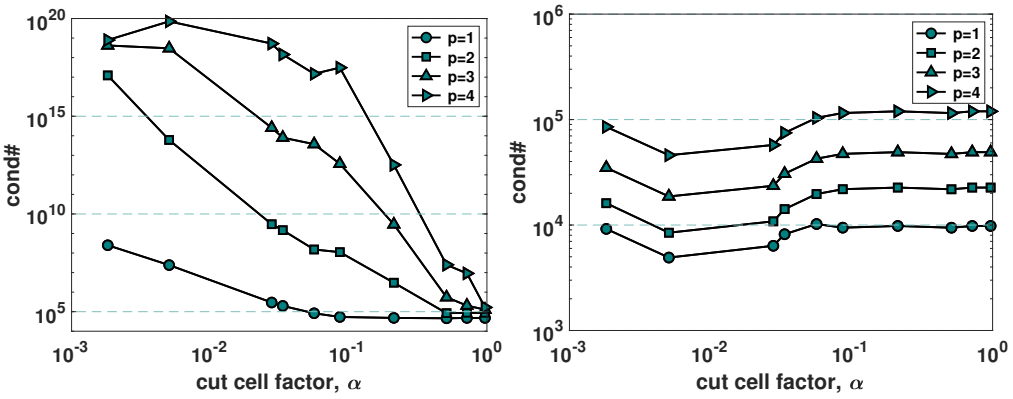
Fig. 4 Cartesian mesh with embedded circle

In this case, the cut-cell factor is given by

$$\alpha = \frac{\text{Area of the smallest cut-cell}}{\text{Area of non-cut cell}}. \tag{9}$$

In this case, we use DG with a Symmetric Interior Penalty Galerkin (SIPG) [27] penalty for the interface flux functions. For the DGD case, we use GD basis, in addition.

Figure 5 compares the condition number of the stiffness matrix for two discretizations. The Figure 5(a) shows the condition number using the DG method for different cut-cell sizes. As observed in the one-dimensional advection case, the condition number also increases significantly as α is decreased. And, again, for the DGD case, the condition number remains almost constant with decreasing α.



(a) DG (b) DGD

Fig. 5 Condition number vs cut-cell factor, α, for a two-dimensional Poisson PDE

The above tests support our hypothesis that DGD will alleviate, if not eliminate, the small-cell problem. As mentioned earlier, the small-cell problem has been addressed previously in several distinct ways. The advantage of the DGD method is that it does not require any special treatment for cut-cells beyond the aforementioned quadrature rules. This makes the DGD method straightforward to implement for complex geometries.

IV. Accuracy studies

This section presents numerical experiments that further explore the DGD cut-cell discretization. In particular, we investigate accuracy in the context of Cartesian adaptive meshes and the Euler equations of gas dynamics. The discretization is implemented using the open source finite element library `mfem` [28] and the quadrature algorithm of [24].

A. Steady isentropic vortex problem

To verify the accuracy of the Euler-equations discretization, we begin by solving the steady isentropic vortex problem using the cut DGD discretization, and we compare the results against those obtained using the conforming-mesh DGD discretization. We chose the two-dimensional isentropic vortex because it has an analytical solution and, thus, is useful for verifying accuracy.

The two-dimensional isentropic vortex is a simple flow consisting of circular streamlines and radially varying density and pressure. The exact solution for the two-dimensional steady vortex problem is defined as

$$\begin{aligned} \rho(r) &= \rho_i \left[1 + \frac{\gamma-1}{2} M_i^2 \left(1 - \frac{r_i^2}{r^2}\right)\right]^{\frac{1}{\gamma-1}}, & u(r,\theta) &= -\rho\sqrt{\frac{\gamma p}{\rho}} M_a \sin\theta, \\ v(r,\theta) &= \rho\sqrt{\frac{\gamma p}{\rho}} M_a \cos\theta, & e(r,\theta) &= \frac{p}{\gamma-1} + \frac{1}{2}\gamma p M_a^2, \end{aligned} \tag{10}$$

where r is the radial polar coordinate, and $r_i = 1$ is the reference radius. The density and Mach number at r_i are given by $\rho_i = 1$ and $M_i = 0.5$, respectively. Here, u, v, e are calculated using the isentropic gas relations and M_a is the local

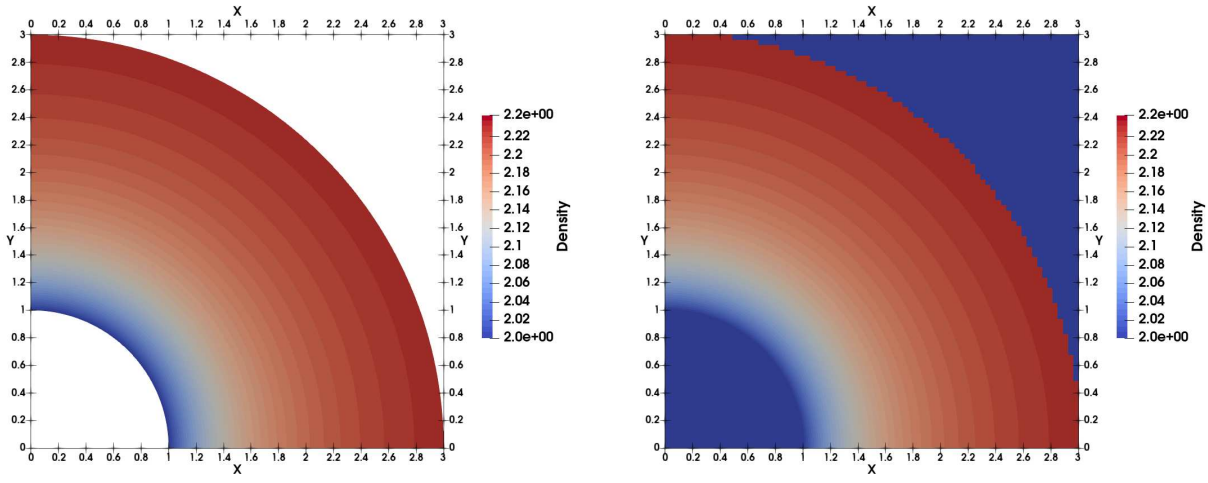
mach number given by

$$M_a = \sqrt{\frac{2}{\gamma - 1}\left[\left(\frac{\rho_i}{\rho}\right)^{\gamma-1}\left(1 + \frac{1}{2}(\gamma - 1)M_i^2\right) - 1\right]}.$$

The domain for the steady-vortex verification is a quarter annulus: $\Omega = \{(r, \theta) \mid 1 \leq r \leq 3,\ 0 \leq \theta \leq \pi/2\}$. The conforming mesh is created by generating an $N \times N$ quadrilateral mesh in polar-coordinate space. For the cut-cell domain case, the background mesh is a simple Cartesian mesh with domain $\{(x, y) \mid 0 \leq x \leq 3,\ 0 \leq y \leq 3\}$, generated using an $N \times N$ quadrilateral mesh in Cartesian space. A slip-wall boundary condition is applied along the inner radius at $r = 1$, and the exact solution is supplied to incoming characteristics on the remaining boundaries. For the slip-wall, the numerical flux is defined by the Euler flux with the normal component of the velocity projected out. Finally, we use the Lax-Friedrichs flux function along interior interfaces.

The solution for $p = 3, N = 80$ is shown in Figure 6 for conforming (non-cut) and cut DGD discretizations. The rough edge for cut-cell DGD solution is a limitation of the plotting software and does not reflect the true domain.

Figure 7 compares the L^2 density error calculated from the two DGD discretizations as a function of element size $h = 1/N$. The results show that degrees $p = 1$ and $p = 3$ have closer to optimal $p + 1$ rates of convergence — both non-cut and cut — while $p = 2$ is sub-optimal. Similar sub-optimal behavior for $p = 2$ was observed in [21].



(a) conforming (non-cut) DGD discretization

(b) cut-cell DGD discretization

Fig. 6 Density solution for $p = 3, N = 80$

B. Flow over an ellipse

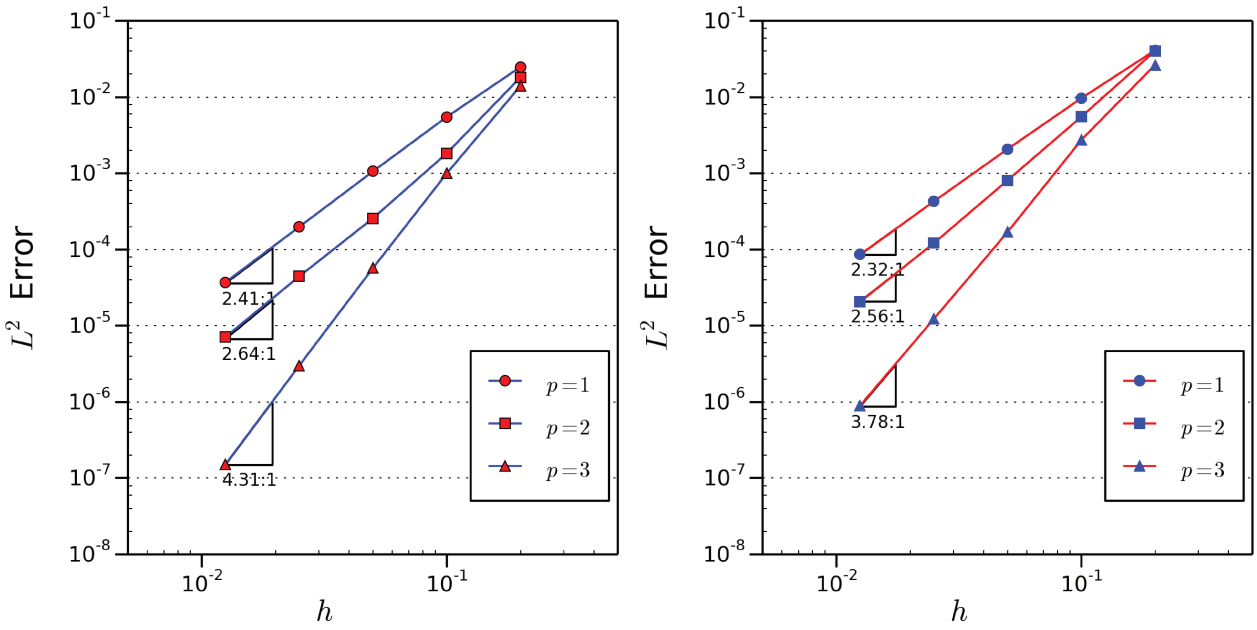
Next we solve the Euler equations to model the flow over an ellipse. We selected the ellipse problem because it approximates an airfoil while having a simple level-set function. We are currently developing a level-set approach for more general geometries.

The ellipse is placed inside a 40×40 square domain and a background mesh is constructed using quadrilateral elements as shown in Figure 8. The ellipse is centered at $(x, y) = (20, 20)$, and its level-set function is

$$\phi(x, y) = \left(\frac{x - 20}{0.5}\right)^2 + \left(\frac{y - 20}{0.05}\right)^2 - 1. \tag{11}$$

The elements in the initial, coarse mesh are refined isotropically (i.e. split into four equal-sized cells) in a non-conforming sense starting from the cells that are cut by the embedded ellipse. The refinement is constrained such that elements have at most two elements along any edge.

The Mach number for the flow is set to $M = 0.5$ and the angle of attack is zero. Along the edges of the square domain, far-field boundary conditions are imposed using a characteristic-based numerical flux function. As with the isentropic vortex, the slip-wall boundary condition at the surface of the ellipse is imposed by removing the normal



(a) conforming (non-cut) DGD discretization

(b) cut-cell DGD discretization

Fig. 7 L^2 density error comparison using DGD for non-cut and cut-cell domain

component of the velocity in the state when evaluating the Euler flux. Unlike the isentropic vortex, we use the Roe numerical flux [29] for the fluxes between interior element interfaces.

We plot density contours for the ellipse problem in Figure 9. Contours are shown for degrees $p = 1, 2$, and 3 DGD discretizations. Density contours closer to the leading and trailing edges are shown separately in Figures 10 and 11, respectively. Qualitatively, the three discretizations produce similar contours, although the higher order results are slightly more symmetric.

For a more quantitative comparison, the drag on the ellipse is evaluated using the same three discretizations and the resulting values are listed in Table 1. For this inviscid and isentropic flow, the analytical value of the drag should be zero. However, numerical dissipation introduces error into the numerical solution, and this is reflected in the drag values in Table 1.

We see that the drag error is roughly equal across the three orders of accuracy, and the high-order methods show no significant advantage; while the drag error for $p = 3$ is half that of $p = 1$, we observe that the error for $p = 2$ is higher than the lowest order scheme. We believe that additional refinement will improve the accuracy of the high-order methods more than the $p = 1$ scheme; unfortunately, the (serial) limitations of our implementation relegate such a study to future work.

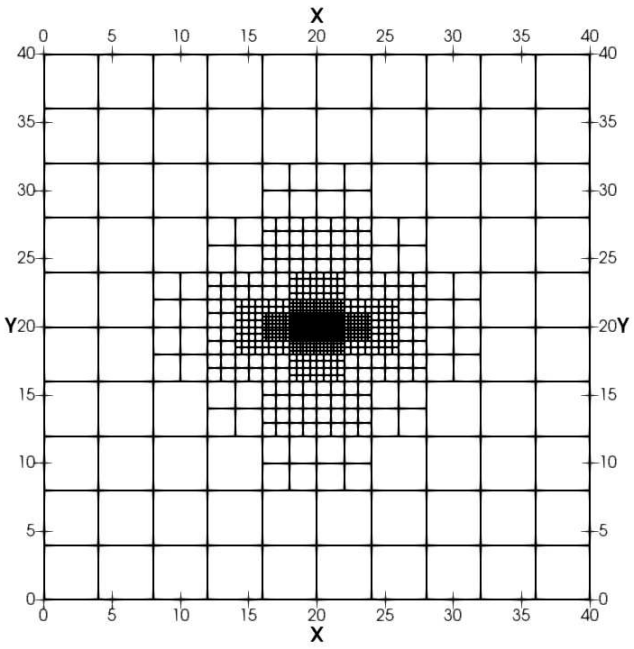
Before concluding this example, we remark that all three degree-p discretizations have the same number of degrees of freedom, unlike DG methods on the same mesh.

Table 1 Drag values for flow over an ellipse

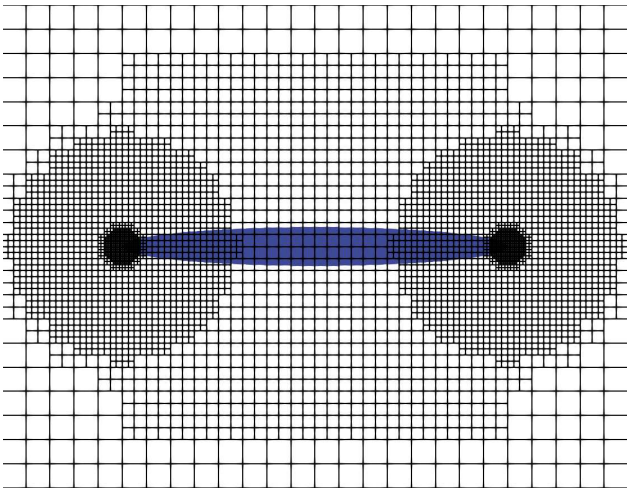
degree (p)	1	2	3
Drag	0.000317462	0.000380608	0.00015232

V. Summary and future work

We have presented a DGD discretization for cut-cell meshes, and we have verified our hypothesis that the DGD stencil helps to alleviate the small-cell problem. Results were presented that verify the accuracy of the discretization in

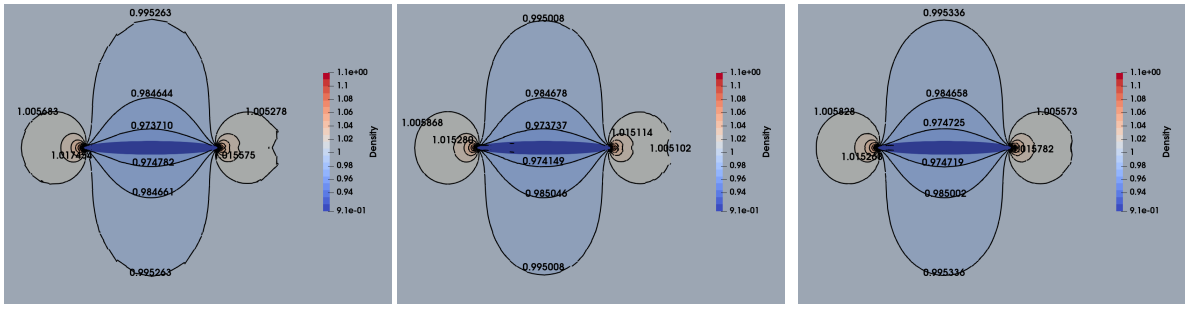


(a) Grid for full domain



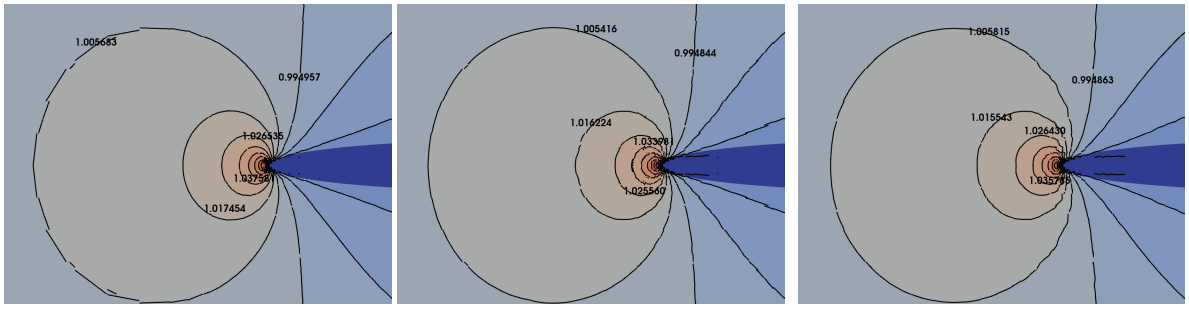
(b) grid closer to embedded geometry

Fig. 8 Background mesh for the ellipse problem



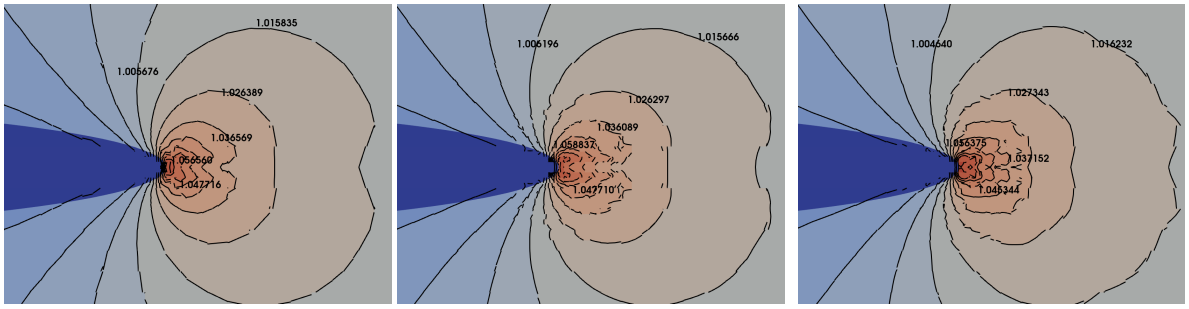
(a) $p = 1$ (b) $p = 2$ (c) $p = 3$

Fig. 9 Density contour plot, full domain



(a) $p = 1$ (b) $p = 2$ (c) $p = 3$

Fig. 10 Density contour plot, leading edge



(a) $p = 1$ (b) $p = 2$ (c) $p = 3$

Fig. 11 Density contour plot, trailing edge

the context of the Euler equations. Future work will address entropy stability and level-set functions for more complex domains. We are also seeking a formal theoretical explanation for the favorable conditioning of the cut-cell DGD method.

Acknowledgments

S. Kaur was supported by the National Science Foundation under Grant No. 1825991. The authors gratefully acknowledge this support. We also thank RPI's Scientific Computation Research Center for the use of computer facilities.

References

[1] Purvis, J. W., and Burkhalter, J. E., "Prediction of critical Mach number for store configurations," *AIAA Journal*, Vol. 17, No. 11, 1979, pp. 1170–1177. https://doi.org/10.2514/3.7617.

[2] Clarke, D. K., Salas, M. D., and Hassan, H. A., "Euler calculations for multielement airfoils using Cartesian grids," *AIAA Journal*, Vol. 24, No. 3, 1986, pp. 353–358. https://doi.org/10.2514/3.9273.

[3] R. Gaffney, J., and Hassan, H., "Euler calculations for wings using Cartesian grids," *Paper 1987-0356, AIAA*, 1987. https://doi.org/10.2514/6.1987-356.

[4] Berger, M., and Leveque, R., "An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries," *9th Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 1989. https://doi.org/10.2514/6.1989-1930, URL http://dx.doi.org/10.2514/6.1989-1930.

[5] Coirier, W. J., and Powell, K. G., "An Accuracy Assessment of Cartesian-Mesh Approaches for the Euler Equations," *Journal of Computational Physics*, Vol. 117, No. 1, 1995, pp. 121 – 131. https://doi.org/https://doi.org/10.1006/jcph.1995.1050, URL http://www.sciencedirect.com/science/article/pii/S0021999185710509.

[6] Aftosmis, M. J., Berger, M. J., and Melton, J. E., "Robust and efficient Cartesian mesh generation for component-based geometry," *AIAA journal*, Vol. 36, No. 6, 1998, pp. 952–960.

[7] Qin, R., and Krivodonova, L., "A discontinuous Galerkin method for solutions of the Euler equations on Cartesian grids with embedded geometries," *Journal of Computational Science*, Vol. 4, No. 1, 2013, pp. 24 – 35. https://doi.org/https://doi.org/10.1016/j.jocs.2012.03.008, URL http://www.sciencedirect.com/science/article/pii/S1877750312000282, computational Methods for Hyperbolic Problems.

[8] Schott, B., and Wall, W., "A new face-oriented stabilized XFEM approach for 2D and 3D incompressible Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 276, 2014, pp. 233–265. https://doi.org/10.1016/j.cma.2014.02.014.

[9] Fidkowski, K. J., and Darmofal, D. L., "A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 225, 2007, pp. 1653–1672.

[10] Sticko, S., and Kreiss, G., "Higher Order Cut Finite Elements for the Wave Equation," *Journal of Scientific Computing*, 2019. https://doi.org/10.1007/s10915-019-01004-2.

[11] Sun, H., and Darmofal, D. L., "An adaptive simplex cut-cell method for high-order discontinuous Galerkin discretizations of elliptic interface problems and conjugate heat transfer problems," *Journal of Computational Physics*, Vol. 278, 2014, pp. 445 – 468. https://doi.org/https://doi.org/10.1016/j.jcp.2014.08.035, URL http://www.sciencedirect.com/science/article/pii/S0021999114006032.

[12] Burman, E., Claus, S., Hansbo, P., Larson, M. G., and Massing, A., "CutFEM: Discretizing geometry and partial differential equations," *International Journal for Numerical Methods in Engineering*, Vol. 104, No. 7, 2015, pp. 472–501. https://doi.org/10.1002/nme.4823, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4823.

[13] Frachon, T., and Zahedi, S., "A cut finite element method for incompressible two-phase Navier-Stokes flows," *Journal of Computational Physics*, Vol. 384, 2019, pp. 77 – 98. https://doi.org/https://doi.org/10.1016/j.jcp.2019.01.028, URL http://www.sciencedirect.com/science/article/pii/S0021999119300798.

[14] Bayyuk, S., Powell, K., and van Leer, B., "A Simulation Technique for 2-D Unsteady Inviscid Flows Around Arbitrarily Moving and Deforming Bodies of Arbitrary Geometry," 1993. https://doi.org/10.2514/6.1993-3391.

[15] Quirk, J. J., "An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies," *Computers Fluids*, Vol. 23, No. 1, 1994, pp. 125 – 142. https://doi.org/https://doi.org/10.1016/0045-7930(94)90031-0, URL http://www.sciencedirect.com/science/article/pii/0045793094900310.

[16] Burman, E., and Hansbo, P., "Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method," *Applied Numerical Mathematics*, Vol. 62, No. 4, 2012, pp. 328 – 341. https://doi.org/https://doi.org/10.1016/j.apnum.2011.01.008, URL http://www.sciencedirect.com/science/article/pii/S0168927411000298, third Chilean Workshop on Numerical Analysis of Partial Differential Equations (WONAPDE 2010).

[17] Burman, E., "Ghost penalty," *Comptes Rendus Mathematique*, Vol. 348, No. 21, 2010, pp. 1217 – 1220. https://doi.org/https://doi.org/10.1016/j.crma.2010.10.006, URL http://www.sciencedirect.com/science/article/pii/S1631073X10002827.

[18] Lehrenfeld, C., and Reusken, A., "Optimal preconditioners for Nitsche-XFEM discretizations of interface problems," *Numerische Mathematik*, Vol. 135, 2017, pp. 313–332.

[19] Hagstrom, T., Banks, J. W., Buckner, B. B., and Juhnke, K., "Discontinuous Galerkin Difference Methods for Symmetric Hyperbolic Systems," *Journal of Scientific Computing*, Vol. 81, No. 3, 2019, pp. 1509–1526.

[20] Li, R., Ming, P., Sun, Z., and Yang, Z., "An Arbitrary-Order Discontinuous Galerkin Method with One Unknown Per Element," *Journal of Scientific Computing*, Vol. 80, No. 1, 2019, pp. 268–288. https://doi.org/10.1007/s10915-019-00937-y, URL https://doi.org/10.1007/s10915-019-00937-y.

[21] Yan, G., Kaur, S., Hicken, J. E., and Banks, J. W., "Entropy-stable Galerkin difference discretization on unstructured grids," *AIAA AVIATION 2020 FORUM*, American Institute of Aeronautics and Astronautics, 2020. https://doi.org/10.2514/6.2020-3033.

[22] Osher, S., and Sethian, J. A., "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, Vol. 79, No. 1, 1988, pp. 12 – 49. https://doi.org/https://doi.org/10.1016/0021-9991(88)90002-2, URL http://www.sciencedirect.com/science/article/pii/0021999188900022.

[23] Fernández-Fidalgo, J., Clain, S., Ramírez, L., Colominas, I., and Nogueira, X., "Very high-order method on immersed curved domains for finite difference schemes with regular Cartesian grids," *Computer Methods in Applied Mechanics and Engineering*, Vol. 360, 2020, p. 112782. https://doi.org/https://doi.org/10.1016/j.cma.2019.112782, URL http://www.sciencedirect.com/science/article/pii/S0045782519306747.

[24] Saye, R., "High-Order Quadrature Methods for Implicitly Defined Surfaces and Volumes in Hyperrectangles," *SIAM Journal on Scientific Computing*, Vol. 37, 2015, pp. A993–A1019. https://doi.org/10.1137/140966290.

[25] de Prenter, F., Verhoosel, C., van Zwieten, G., and van Brummelen, E., "Condition number analysis and preconditioning of the finite cell method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 316, 2017, pp. 297 – 327. https://doi.org/https://doi.org/10.1016/j.cma.2016.07.006, URL http://www.sciencedirect.com/science/article/pii/S0045782516307277, special Issue on Isogeometric Analysis: Progress and Challenges.

[26] Banks, J., and Hagstrom, T., "On Galerkin Difference Methods," *Journal of Computational Physics*, Vol. 313, 2016. https://doi.org/10.1016/j.jcp.2016.02.042.

[27] Arnold, D., and Douglas, N., "An Interior Penalty Finite Element Method with Discontinuous Elements," *SIAM Journal on Numerical Analysis*, Vol. 19, 1982. https://doi.org/10.1137/0719052.

[28] Kolev, T., and Dobrev, V., "Modular Finite Element Methods (MFEM)," [Computer Software] https://doi.org/10.11578/dc.20171025.1248, jun 2010. URL https://doi.org/10.11578/dc.20171025.1248.

[29] Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372. https://doi.org/10.1016/0021-9991(81)90128-5, URL http://dx.doi.org/10.1016/0021-9991(81)90128-5.